SIMULASI BYTE ERROR CORRECTION DENGAN MENGGUNAKAN HAMMING CODE

Mahyuzar Masri[1)], Hermansyah Alam[2)], Mahrizal Masri[3)], Zulkarnain Lubis[4)], Muhammad Izhni Pohan[5)]

[1,2,3,4)]Dosen Teknik Informatika – ITM, [5)]Alumni Teknik Informatika - ITM

Abstrak

Penggunaan komputer dalam memproses data digital tidak terlepas dari fungsi RAM yang merupakan tempat penyimpanan memori dari komputer tersebut.Seperti layaknya komponen elektronik lainnya, seiring dengan jumlah pemakaian komputer, RAM akan mengalami penurunan tingkat kemampuan yang menyebabkan dapat terjadinya *error* atau kesalahan dalam proses pembacaan, transmisi maupun penyimpanan data digital. Sebuah metode yang dapat mendeteksi atau bahkan memperbaiki *error* yang terjadi pada *parity bit* data digital sangat bermanfaat dalam hal mengatasi masalah tersebut. *Hamming code* merupakan metode yang dapat mendeteksi sekaligus melakukan perbaikan kesalahan pada *parity bit* data dengan cara melakukan pengecekan terhadap struktur bit data yang bersangkutan. Metode *hamming code* ini diimplementasi ke dalam sebuah bentuk simulasi, sehingga menghasilkan simulasi cara kerja *hamming code* yang menunjukkan bagaimana metode ini melakukan *bit error correction* atau perbaikan kesalahan pada bit data digital. Simulasi ini dirancang menggunakan bahasa pemrograman *Microsoft Visual Studio 2010*, dengan bentuk tampilan langkah demi langkah proses pengecekan kesalahan dan perbaikan kesalahan berdasarkan cara kerja metode *hamming code* yang digunakan.

Kata Kunci: *Algoritma, Byte Error Correction, Hamming Code, Visual Studio 2010*

I. PENDAHULUAN

Komputer merupakan perangkat elektronik yang terdiri dari rangkaian transistor dankapasitor yang membentuk sebuah rangkaian dengan fungsi yang berbeda-beda. Penggunaan komputer dalam memproses data digital tidak terlepas dari fungsi RAM yang merupakan tempat penyimpanan memori dari komputer tersebut. Sebuah RAM dapat terdiri dari kumpulan transistor dan kapasitor yang tersusun secara rapat, sehingga mampu menyimpan informasi data digital di dalamnya.

Seperti layaknya komponen elektronik lainnya, seiring dengan jumlah pemakaian komputer, transistor dan kapasitor yang ada di dalamnya akan mengalami penurunan tingkat kemampuan. Hal ini menyebabkan dapat terjadinya *error* atau kesalahan dalam proses pembacaan, transmisi maupun penyimpanan data digital. *Error* ini terjadi pada *parity bit* dari data yang tersimpan, dimana nilainya berubah dari nilai sebelumnya, sehingga menyebabkan data tersebut rusak atau bahkan tidak dapat diakses sama sekali.

Untuk mengatasi masalah ini, perlu adanya sebuah metode yang dapat mendeteksi atau bahkan memperbaiki *error* yang terjadi pada *parity bit* tersebut. Salah satu metode yang yang dapat digunakan untuk mendeteksi atau bahkan memperbaiki *error* yang terjadi pada *parity bit* data adalah *hamming code*. Metode ini menambahkan sejumlah n bit pada *parity bit* data, sehingga terjadi perubahan data dengan jumlah bit yang bertambah.Data yang mengalamiperubahan inilah yang nantinya akan diproses atau disimpan melalui RAM. Penambahan n bit pada data ini berfungsi sebagai kode, yang nantinya digunakan dalam pengecekan apakah data yang diproses atau disimpan mengalami *error*. Jika *error* terjadi pada data yang diproses atau disimpan, n bit yang ditambahkan juga berfungsi sebagai kode untuk memperbaiki kesalahan pada data tersebut.

II. TINJAUAN PUSTAKA

2.1 Pengertian Data

Data merupakan salah satu hal utama yang dibahas dalam Teknologi Informasi Komputer. Penggunaan dan pemanfaatan data sudah mencakup banyak aspek.Data biasanya terdiri dari beberapa elemen data (dataitem). Elemen dataadalah unit terkecil dari data yang ada artinya bagi pihak yang menggunakannya(user). Dalam suatu sistem basis data, elemen data ini disebut dengan Field (Setiawan &Munir : 2006).

2.2 Error Detection And Correction.

Konsep dari *error delection and correction* adalah untuk mengurangi dampak kesalahan (error) yang bisa diakibatkan oleh masalah pengamatandari informasi digital terhadap kesalahan yang kerap muncul karena pada saat data proses transmisi tersebut, justru mengakibatkan sistem mejadi salah pengertian melakukan interpretasi terhadap simbol data-data yang akan diterjemahkan untuk kemudian diproses menjadi sebuah pesan.adapun beberapametode yang dilakukan adalah 9 Mano & Kime : 2003):

1. ***Backward Error Control* (BEC)**

Metode dimana perangkat sebuah proses pengiriman akan segera mengirimkan sinyal kepada perangkat pengirim untuk melakukan pengiriman sinyal kepada perangkat pingiriman untuk melakukan pengiriman ulang jika pada data yang diterimah terjadi kesalahan.

2. *Forward Error Control* (FEC)

Metode dimana sebelum proses pengiriman data dilakukan terlebih dahulu dikodekan dengan suatu pembangkit kode (encoder), kemudian dikirim keperangkat penerimah.

2.3 Algoritma

Algoritma adalah prosedur komputasi yang didefenisi dengan baik yang mengambil beberapa nilai yaitu seperangkat nilai sebagai input dan output yang menghasilkan nilai.

2.4 Hamming Code

Teknik pengkodean Hamming memiliki beberapa keunggulan dimana dapat tepat mengkoreksi satu kesatuan kesalahan bit yang timbul. Dalam proses pengkodean kode hamming diperlukan suatu generator matriks untuk mengubah sejumlah susunan bit streamdata yang diterimah sebelum dilakukan pengkodean kembali untuk kemudian dikirimkan ke sisi pertama.

2.5 UML (Unified Modeling Language).

UML (Unified Modeling Language) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah meteologi untuk mengmbangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut.langkah kerja UML dalam merancang sebuah program software yang di sertakan contoh dari sebuah aplikasi,UML meggunakan langkah UseCase, Aktor, Dentifikasi Use Case, Pendokumentasian Model UseCase. *Activity Diagram* (Diagram Aktivitas) merupakan diagram flowchart yang disempurnakan. Diagram aktifitas menggambarkan operasi pada suatu obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram *flowchat*adalah adanya dukungan konkurensi (pelaksanaan aktivitas secara bersamaan), pengiriman pesaan dan *swimlane* (pelaku/penanggung jawab aktivitas).

2.6 Microsoft Visual Studio 2010

Microsoft visual basic (sering disingkat sebagai VB saja) merupakan sebuah bahasa pemograman yang bersifat event drive dan menawarkan *intregrited devolopment* (IDE) visual untuk membuat program aplikasi berbasis sistem operasi Microsoft Windows dengan menggunakan model pemograman Common Object Model (COM) (Sianipar,2014).

III. METODE PENELITIAN

3.1 Tempat dan Jadwal Penelitian

Penelitian ini dilaksanakan dengan melakukan observasi dan studi terhadap yang berhubungan dengan permasalahan pengecekan *byte error* dengan menggunakan metode *hamming code*. Oberservasi dan studi yang dilakukan dilaksanakan pada perpustakaan Institut Teknologi Medan.

Tabel 1. Jadwal Penelitian

Kegiatan	Bulan Pelaksanaan															
	Juni			Juli				Agustus				September				Oktober
Studi Literatur																
Pengumpulan Data																
Analisis dan Perancangan																
Implementasi																
Pengujian																

3.2 Implementasian

Metode ini dilaksanakan dengan melakukan studi keputusan-keputusan melalui hasil penelitian lainnya maupun artikel-artikel yang relevan, serta mempelajari lebih dalam teori-teori tentang metode simulasi, *byte error correction* dan metode *hamming code*.

3.3 Analisa Sistem

Proses pembentukan *parity bit* dilakukan agar data yang ditransfer dapat dicek tingkat kesalahannya pada proses pengiriman serta dapat dilakukan perbaikan jika ditemukan kesalahan pada data tersebut. Pembentukan *parity bit* ini dilakukan dengan menambahkan beberapa bit tambahan pada data yang akan di transfer. Penambahan ini dilakukan pada posisi bit kelipatan 2 (dua) dari deret data yang akan ditransfer.

Sebagai contoh, akan dilakukan transfer terhadap data dengan nilai bit: **1011011**. Dari contoh ini terlihat bahwa ada 7 bit data yang akan ditransfer. Langkah pertama dari pembentukan *parity bit* adalah mempersiapkan posisi *parity* yang nantinya akan diisi, sebagaimana terlihat pada Gambar 1.

Posisi Bit	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1
Nilai	1	1	0	-	1	1	0	-	1	-	-

Gambar 1. Persiapan *Parity Bit*

Selanjutnya, akan ditentukan nilai *parity bit* yang akan disisipkan ke dalam posisi *parity* yang sudah disiapkan, yaitu pada posisi R1, R2, R4 dan R8.

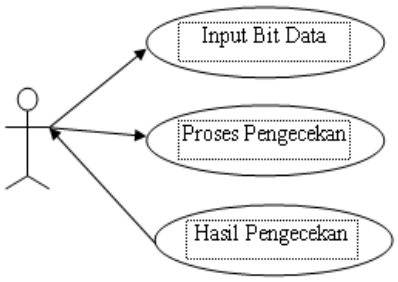
Untuk posisi R1, di tentukan terlebih dahulu posisi bit mana saja yang memiliki nilai 1 pada posisi paling ujung dari *least significant bit*-nya. R1, R3, R5, R7, R9 dan R11 merupakan posisi yang jika diubah menjadi nilai bit akan memiliki nilai.

3.4 Perancangan Sistem

Perancangan aplikasi meliputi perancangan *Unified Modelling Language* (*usecase diagram* dan *activity diagram*) dan antarmuka yang dibutuhkan.

1. *Usecase Diagram* Sistem

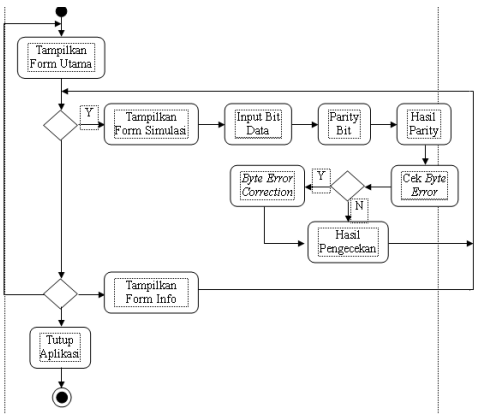
Usecase diagram dirancang untuk menjelaskan bentuk rancangan interaksi yang bisa dilakukan antara pengguna dengan perangkat lunak simulasi *byte error correction* dengan menggunakan metode *hamming coding*.



Gambar 2. UseCase Diagram Sistem

2. *Activity Diagram* Sistem

Activity diagram dirancang untuk menjelaskan bentuk rancangan proses aktivitas yang bisa dilakukan oleh pengguna melalui perangkat lunak simulasi *byte error correction* dengan metode *hamming code* yang dirancang.



Gambar 3. Activity Diagaram Sistem

IV. HASIL DAN PEMBAHASAN

4.1 Pembahasan

Berdasarkan hasil implementasi sistem dan pengujian sistem sebelumnya, pembahasan yang dilakukan terhadap sistem yang dihasilkan mencakup kelebihan dan kelemahan sistem.

- Kelebihan Sistem

Berdasarkan hasil implementasi metode *hamming code* pada aplikasi simulasi *byte error correction* ini, beberapa kelebihan dari sistem yang dihasilkan adalah sebagai berikut :

1. Sistem dapat menampilkan proses pembentukan parity bit yang digunakan dalam metode *hamming code* untuk pengecekan dan perbaikan kesalahan bit pada transfer data.
2. Selain dapat menerima inputan nilai bit yang akan ditransfer dari pengguna, sistem juga dapat menerima perubahan nilai parity bit sesuai dengan keinginan pengguna.
3. Sistem dapat menampilkan rincian proses pengecekan parity bit hasil transfer data, menunjukkan posisi bit yang mengalami kesalahan serta solusi untuk memperbaiki kesalahan tersebut, sesuai dengan metode *hamming code* yang digunakan dalam penelitian ini.
    - Kelemahan Sistem
      Adapun kelemahan dari sistem yang ditemukan adalah sebagai berikut :
    - Sistem belum mampu untuk menerima nilai bit yang melebihi 8 bit. Dengan 8 bit inputan nilai data dan 4 bit untuk redundant bit, total 12 bit merupakan maksimum bit yang dapat diproses oleh sistem.
    - Sistem hanya dapat mengecek dan mempperbaiki kesalahan yang terjadi pada perubahan nilai bit untuk 1 posisi bit saja. Jika perubahan nilai bit terjadi pada lebih dari satu posisi bit, sistem sudah tidak dapat memperbaiki kesalahan tersebut.

4.2 Pengujian Aplikasi

Setelah memperoleh hasil dari perancangan sebelumnya, yaitu berupa sebuah aplikasi simulasi *byte error correction* dengan metode *hamming code*, dilakukan pengujian serta pembahasan mengenai hasil yang dapat diperoleh.

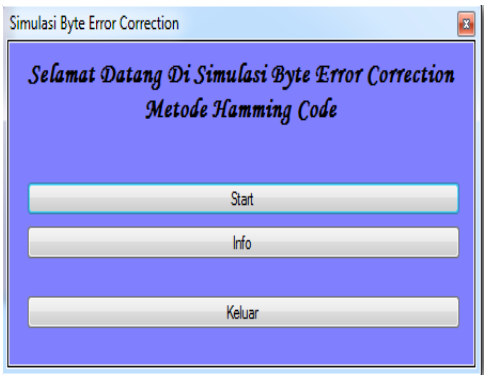
Hasil dari perancangan aplikasi simulasi *byte error correction* dengan metode *hamming code* ini diuji untuk melihat apakah hasilnya telah sesuai dengan rencana awal penelitian. Adapun tahapan pengujian yang dilakukan adalah:

1. Menyiapkan Perangkat Pengujian

Dalam tahap ini, penulis melakukan pengujian transfer data dengan nilai bit: **1011001**.
Setelah transfer data selesai, data yang diterima akan diubah pada salah satu posisi bitnya.

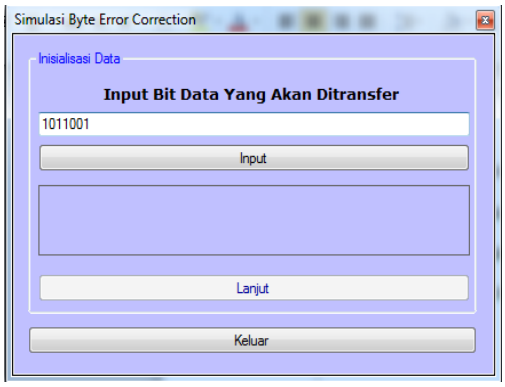
2. Pengujian *Form* Utama

Pengujian *form* Utama dilakukan untuk menguji apakah simulasi sudah dapat dieksekusi dengan baik. Untuk itu, dilakukan eksekusi terhadap aplikasi simulasi ini sehingga muncul tampilan *form* Utama sebagaimana terlihat pada Gambar 4.



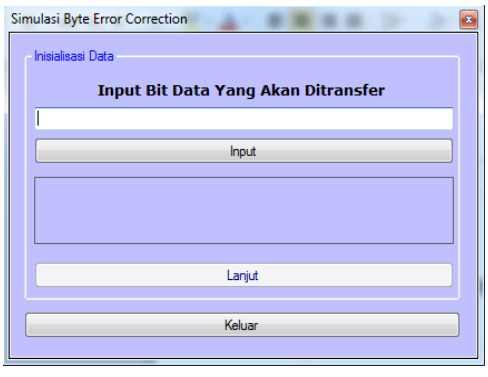
Gambar 4. Tampilan Form Utama

Selanjutnya dilakukan penekanan tombol Start, yang menghasilkan tampilan awal form Simulasi, seperti terlihat pada Gambar 5.



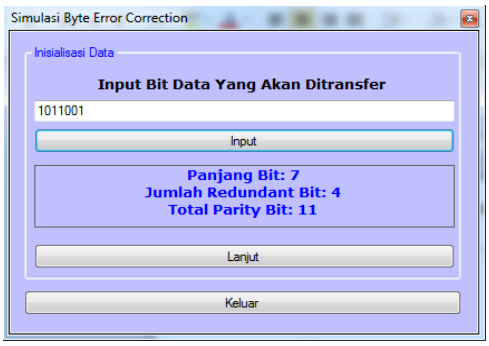
Gambar 5. Tampilan Awal Form Simulasi

Langkah berikutnya adalah menginputkan nilai bit data yang akan ditransfer. Sesuai dengan data yang sudah disiapkan pada persiapan pengujian sebelumnya, diinputkan nilai bit **1011001** pada textbox yang tersedia, sebagaimana terlihat pada Gambar 6.



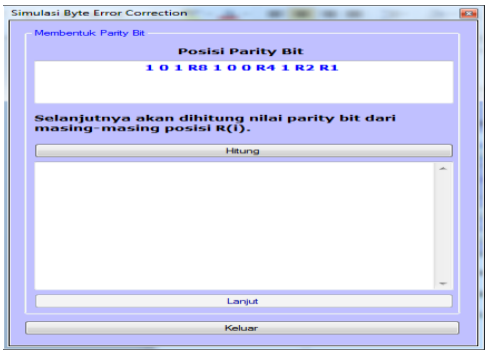
Gambar 6. Tampilan Input Bit Data

Kemudian dilakukan penekanan tombol Input, yang menghasilkan informasi mengenai nilai bit data yang diinputkan, sebagaimana terlihat pada Gambar 7.



Gambar 7. Hasil Tombol Input

Kemudian dilakukan penekanan tombol Lanjut, yang menghasilkan munculnya tampilan pembentukan parity bit, seperti terlihat pada Gambar 8.

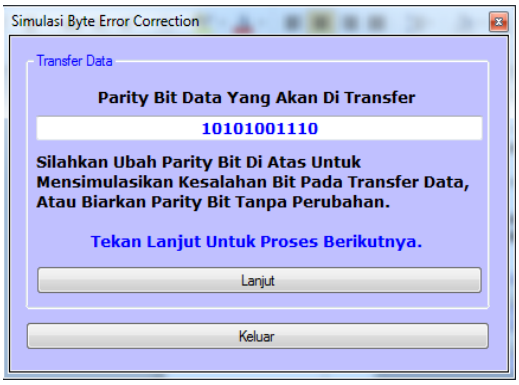


Gambar 8. Tampilan Pembentukan Parity Bit

Langkah berikutnya adalah penekanan tombol hitung, yang menghasilkan rincian nilai parity bit untuk masing-masing posisi *redundant bit*, seperti terlihat pada Gambar 9.

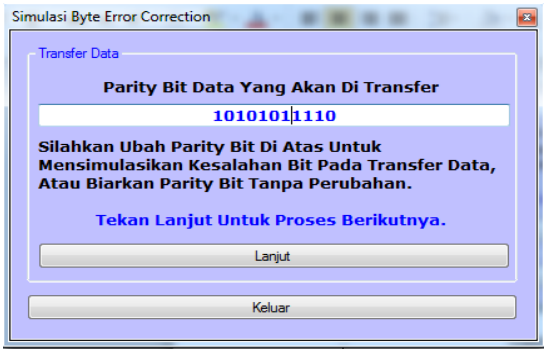
Gambar 9. Hasil Perhitungan Parity Bit

Selanjutnya dilakukan penekanan tombol Lanjut, yang menghasilkan munculnya tampilan transfer data, sebagaimana terlihat pada Gambar 10.



Gambar 10. Tampilan Transfer Data

Dari Gambar 10, terlihat bahwa parity bit data yang akan di transfer adalah **10101001110**. Untuk menguji apakah simulasi sudah dapat melakukan pengecekan dan perbaikan terhadap kesalahan bit pada transfer data, nilai parity bit tersebut akan diubah salah satu posisi bitnya, sehingga berubah menjadi **10101011110**. Nilai ini kemudian diinputkan pada textbox yang tersedia, sehingga menghasilkan tampilan seperti terlihat pada Gambar 11.



Gambar 11. Tampilan Perubahan Parity Bit Data

Kemudian dilakukan penekanan tombol lanjut, sehingga muncul tampilan parity bit yang diterima seperti terlihat pada Gambar 12.

Gambar 12. Tampilan Parity Bit Yang Diterima

Langkah berikutnya adalah melakukan penekanan tombol cek yang menghasilkan tampilan proses pengecekan *error* dan hasil perbaikannya, seperti terlihat pada Gambar 13.

Gambar 13. Tampilan Hasil Pengecekan *Error*

Selanjutnya dilakukan penekanan tombol Keluar, sehingga tampilan form Utama kembali terlihat, yang mana selanjutnya akan dilakukan pengujian terhadap form Info

1. Pengujian *Form* Info

Pengujian *form* Info dilakukan untuk menguji apakah sistem sudah dapat menampilkan informasi perancang aplikasi dengan baik. Untuk itu, dilakukan penekanan tombol Info pada *form* Simulasi, sehingga muncul tampilan *form* Info sebagaimana terlihat pada Gambar 14.

Gambar 14. Tampilan Form Info

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian yang dilakukan terhadap perancangan aplikasi simulasi *byte error correction* dengan metode *hamming code* ini, penulis dapat menarikbeberapa kesimpulan sebagai berikut

1. Metode *hamming code* dapat digunakan untuk mendeteksi kesalahan yang terjadi pada bit data pada saat proses transfer data berlangsung. Dengan memanfaatkan nilai parity bit yang dibentuk sebelumnya, metode ini mampu mendeteksi apakah data mengalami perubahan bit pada saat proses transfer data selesai.
2. Metode *hamming code* juga mampu memperbaiki kesalahan pada proses transfer data, dimana terjadi perubahan nilai bit pada data yang ditransfer.
3. Dengan mengimplementasikan sebuah metode ke dalam aplikasi simulasi diperoleh sebuah aplikasi yang dapat digunakan untuk memahami bagaimana cara kerja metode tersebut.

5.2 Saran

Untuk pengembangan penelitan selanjutnya, ada beberapa saran yang ingin penulis berikan, berhubungan dengan hasil dari penelitian ini. Adapun saran-saran tersebut adalah sebagai berikut:

1. Aplikasi ini dapat dikembangkan sehingga dapat mendeteksi dan memperbaiki kesalahan bit pada data yang memiliki jumlah bit lebih dari 8.
2. Aplikasi ini dapat dikembangkan dengan menambahkan metode pengecekan kesalahan bit lainnya, sehingga dapat dilihat perbedaannya dengan metode *hamming code* yang digunakan dalam penelitian ini.

DAFTAR PUSTAKA

[1.] Gupta, B. K. & Sinha, R., 2013, *Novel Hamming Code For Error Correction And Detection Of Higher Data Bits Using VHDL*, International Journal of Scientific & Engineering Research, Volume 4, Issue 4, ISSN 2229-5518

[2.] Ladjamudin, A. B. 2013, *Analisis dan Desain Sistem Informasi*, Graha Ilmu. Yogyakarta.

[3.] Mano, M.M & Kime, C., 2003, *Logic and Computer Design Fundamentals*, Edisi Ketiga, Prentice Hall, New Jersey.

[4.] Munir, R., 2007, *Algoritma dan Pemrograman*, Informatika Bandung.

[5.] Proakis,J. G., 2001, *Digital Communication*, McGraw Hill, Singapore.

[6.] Riyanto,M. Z., 2004, *Komunikasi Data*, Jurnal UGM, Yogyakarta.

[7.] Setiawan, W., & Munir, 2006, *Pengantar Teknologi Informasi* : Basis Data, Universitas Pendidikan Indonesia, Bandung.

[8.] Sianipar, R.H., 2014, *Pemrograman Visual Basic.* NET, Informatika, Bandung.

[9.] Zarlis, M. & Handrizal, 2008, *Teori Pemrograman dan Praktik Dalam Pascal*, Edisi Kedua,USU Press, Medan